

## **REMARKS**

Applicants respectfully request reconsideration of the present application.

### **Rejections based on 35 U.S.C. § 102**

#### **I. Applicable Authority**

“A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.” *Verdeggal Brothers v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). “The identical invention must be shown in as complete detail as is contained in the . . . claim.” *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). *See also*, MPEP § 2131.

#### **II. The Rejection under 35 U.S.C. § 102(b) over Fedorov et al. (Professional Active Server Pages 2.0,) 1998, Wrox Press Ltd) should be withdrawn because Fedorov fails to disclose, among other things, “generating a host wizard having one or more transitional pages, wherein the host wizard defines an extension interface to respond to navigation events; and generating a user interface that integrates said web component into said host wizard by utilizing the extension interface to perform recursive navigation between said web component and said host wizard.”**

Claims 7, 10, and 11 stand rejected under 35 U.S.C. 102(b) as being anticipated by Alex Fedorov, et al. “Professional Active Server Pages 2.0”, 1998, Wrox Press Ltd (hereinafter Fedorov, et al.). Applicant respectfully traverses this rejection as follows.

Applicants submit that the Office’s interpretation of “wizard” as utilized in claims 7, 10, and 11 and defined in the specification, at p. 2, ll. 7-10, goes beyond the reasonable ordinary and customary meaning of the term as understood by an artisan of ordinary skill. *Philips v. AWH Corp.*, 415 F.3d 1303, 1313, 75 USPQ2d 1321, 1326 (Fed. Cir. 2005). In *Philips*, the court noted “[i]mportantly, the person of ordinary skill in the art is deemed to read

the claim term not only in the context of the particular claim in which the disputed term appears but in the context of the entire patent, including the specification. Id. at 1313.

In the specification, at p. 2, ll. 7-10, Applicants define “wizard” to mean an application that directs a user through a configuration process or the implementation of a particular task within an application program; a “wizard” is essentially a programmatic method of providing guidance to a user within a controlled environment and in a predictable manner. The programmatic method includes a multistep process that is controlled by a user’s navigation of screens to answer questions and ultimately complete an operation.

Applicants’ specification at p. 5, ll. 15-24 and page 11, ll. 15-20, further teaches wizards provided by the invention enable reuse, extensibility, and information sharing among host-wizards and sub-wizards. Applicants have requested the Office to provide a reasonable interpretation of wizard as in the context of the pending claims and Applicants’ specification.

The Office’s conclusion that a host wizard is an Active Server Page (seismic.asp) is unreasonable. *See, e.g.*, Final Office Action mailed 08/25/10 at p. 3. The Office’s interpretation of host-wizard is beyond the ordinary and customary meaning of the term as utilized in claims 7, 10, and 11 and Applicants’ specification. As taught by Fedorov, at page 423, ll. 30-37, the Active Server Page (seismic.asp) **only** provides server-side calculations, checks a uniform resource locator (URL) for data to process, and loads a “wizard dialog box.” A reasonable reading of Fedorov does not support the Office’s interpretation that seismic.asp is a host-wizard. Seismic.asp is not a wizard that defines an extension interface to respond to navigation events, where the extension interface enables recursive navigation between the host-wizard and a web component. Seismic.asp is not a multistep process that is controlled by a

user's navigation of screens. Rather, seismic.asp merely loads a wizard dialog box that collects data, which is processed by a server executing seismic.asp.

The Office's conclusion that a user interface that integrates a web component into a host wizard by utilizing the extension interface to perform recursive navigation between the web component and host wizard as recited in independent claims 7, 10, and 11 is anticipated by an html page (equakeget.htm) as in Fedorov is also unreasonable. *See, e.g.*, Final Office Action mailed 12/05/06 at p. 3. Fedorov, at page 423, ll. 25-32, teaches the user interface for the wizard dialog box loaded by seismic.asp is provided by equakeget.htm. Equakeget.htm generates the user interface for the wizard dialog box. Equakeget.htm does not integrate a web component into the host wizard by utilizing an extension interface provided by the host wizard to perform recursive navigation between the web component and the host wizard. At best, equakeget.htm only provides a user interface based on navigation within the wizard dialog box. Nothing in Fedorov teaches a user interface based on navigation between a host wizard and a web component provided by an extension interface of the host wizard.

Additionally, the Office reasons that the "Back" and "Next" of the wizard dialog box controls in Fedorov provide recursive navigation between a host wizard and web component. Contrary to the Office's allegation, Fedorov, at page 424, expressly indicates that the "Back" and "Next" controls are used to navigate between panes of the dialog box wizard. Navigation within a wizard is not the same or identical to navigation between wizards. *See, Applicant's specification at pp. 12-16.* Applicant precisely describes the additional coding and interactions that are necessary to enable this novel functionality. Fedorov's description of a prior art wizard without any extension capabilities as expressly claimed does not fairly describe or anticipated the invention of the claims. Nothing in Fedorov, teaches that the "Back" and "Next"

controls provide recursive navigation between a host wizard and a web component. Applicants' argument is not directed to navigation between pages. Rather, Applicants' argument focuses on navigation between wizards.

Fedorov does not describe the identical invention of the claims. Unlike Fedorov, the invention of independent claims 7, 10, and 11, require, among other things, a host-wizard having transitional pages, wherein the host wizard defines an extension interface that responds to navigation events to provide recursive navigation between a web component and the host wizard. Fedorov fails to teach a host wizard and integration of the host wizard and a web component in a manner that provides recursive navigation between the host wizard and the web component when generating the user interface. Accordingly, for at least the reasons set forth above, Applicant respectfully requests withdrawal of the anticipation rejection and allowance of independent claims 7, 10, and 11.

**III. The Rejection under 35 U.S.C. § 102(e) over Gauthier et al (U.S. Patent No. 6,574,791) should be withdrawn because Gauthier fails to teach, among other things, "invoking said one or more sub-wizard components during said host-wizard component execution; and transferring control from said host-wizard to said one or more sub-wizard components."**

Claims 1-3, 8, 9, and 14 stand rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,574,791 to Gauthier, et al.

The Office indicates that Gauthier, at col. 9, l. 56-col. 10, l. 5, anticipates the claimed requirements. The section cited by the office action discloses:

"The WizardManager class defines objects which control the execution of multiple subwizards within the target wizards. This class also references the GUI elements making up the overall target wizard. This class preferably includes constructor methods for creating a new WizardManagerFrame object, a new WizardManagerButtonPanel object, a new WizardManagerSelectionPanel object, and a new WizardManagerLogoPanel object, and constructor methods that creating these objects from specified existing meta data. These constructor methods would call corresponding constructor methods on the

corresponding class. This class also preferably includes an AddWizard( ) method for adding new subwizards to the overall wizard skeleton. This class also preferably includes methods for launching the overall wizard, and launching selected subwizards. The class also preferably includes a describe content method used to export meta data from the class objects.”

The Office reads the first sentence of this citation without the aid of context of the patent. Applicant respectfully reminds the Office that the identical invention must be described by Gauthier. MPEP § 2131.

**a. Independent claims 1, 8, and 9 require, among other things, transfer of control from the host wizard to the subwizard.**

The Office contends that control must be passed to the sub-wizard. However, a reasonable reading of Gauthier, at col. 9, ll. 55-58 and col. 13, ll. 45-50, would lead one of ordinary skill in the art to conclude that control is maintained by the Wizard Manager Class. Thus, contrary to Examiner assertion that passing control is necessary for subwizards to execute, Gauthier expressly states otherwise and explains that control is centralized at the Wizard Manager.

Gauthier describes a “WizardManager” that controls the execution of the subwizards which define the Target wizard, at col. 13, ll. 25-35. The execution of the subwizards is centralized and coordinated by the WizardManager. Gauthier, at col. 13, ll. 45-50, expressly states that “the Wizard Manager object controls execution of the one or more subwizards. The Gauthier disclosure differs from the invention defined by claims 1, 8, and 9 because the control functions of Gauthier are centralized in Wizard Manager and not distributed to the subwizard components as required by claim 1, 8, and 9. The Office further contends that Gauthier, at col. 14, l. 9 through col. 15, l. 24, teaches the claimed “transferring of control to the subwizard.” This is not a fair or reasonable reading of Gauthier as the context of the patent as

whole expressly states the exact opposite of what the Office claims the reference stands for. Gauthier discloses WizardState objects that are utilized to generate code that implement the wizard functions. Nothing in the cited section discloses control is passed to the sub-wizard during the execution of the host wizard.

Unlike Gauthier, independent claims 1, 8, and 9 require, among other things, a host-wizard to transfer control directly to a sub-wizard, and the sub-wizard to control its own execution. Gauthier fails to teach a transfer of control to one or more sub-wizard components. Accordingly, for at least the reasons set forth above, Applicants respectfully request withdrawal of the anticipation rejection and allowance of independent claims 1, 8, and 9.

**b. Dependent claim 2 requires, among other things, subwizard components that are operating system based application components.**

Claim 2 depends from claim 1 and further defines novel features of the claimed invention. Accordingly, claim 2 is allowable by virtue of its dependence on claim 1. Additionally, claim 2 is allowable because Gauthier fails to teach “wherein one or more sub-wizard components are web browser based object components.” The Final Office Action has referenced Gauthier, at col. 18, l. 66-col. 19, l. 30, to anticipate the claimed requirement. Final Office Action at pg. 9. The section cited by the Office discloses:

“Turning now to FIG. 6, the preferred embodiment implementation of the WizardMetaDataManager 128 is illustrated in more detail. As discussed above, the WizardMetaDataManager 128 is used to persist and retrieve target wizard meta data. In the preferred embodiment, the WizardMetaDataManager 128 implementation includes a WizardMetaDataManager object, a WizardMetaData object, a WizardMetaDataFormDescription object, a WizardMetaDataPanel object, and a WizardMetaDataFrame object.

The WizardMetaDataManager object preferably includes a ReuseExistingMetaData( ) method an[d] a SaveWizardMetaData( ). The SaveWizardMetaData( ) displays the WizardMetaDataPanel object which prompts the developer to select the meta data to be stored and specify a wizard meta data file in a form specified by the WizardMetaDataFormDescription object. The ReuseExistingMetaData( ) method displays the

WizardMetaDataPanel and prompts the user to select a set of wizard meta data. The method retrieves the wizard meta data and walks through it to recreate an **internal** set of wizard framework objects.

The WizardMetaData object includes the actual meta data for the target wizard. This would preferably include all of the components of the target wizard and their interrelationships. The storage would preferably be done in a **language neutral format** to facilitate ease of restoring. The WizardMetaDataFormDescription object describes the format used for storing the wizard meta data. For example, the wizard meta[1] data could be described using extensible markup language (XML) and an accompanying wizard framework specific document content description. The WizardMetaDataPanel object and a WizardMetaDataFrame object provide the GUI interface to the developer.”

Gauthier teaches a WizardMetaDataManager that is able to retrieve stored meta data, the meta data preferably being stored in a language neutral format. Gauthier further discloses that it is possible to describe the meta data using XML and a specific document content description. XML is an example of the language neutral format that aids in defining the meta data. Gauthier, at col. 14, ll. 9-12, further discloses that each sub-wizard includes a Wizard object, a WizardState object, a WizardDefault object, a WizardStateController object, WizardPanel object(s), and a WizardCodeGenerator object. Gauthier discloses that the XML format is utilized to represent metadata associated with a target wizard. Gauthier does not teach that XML is utilized to define a target wizard. As described in Gauthier, XML is a storage format and not a web browser based object component that receives control from a host wizard during the execution of the host wizard. Accordingly, Gauthier fails to disclose sub-wizards that include browser based object components. Applicants' specification, at page 4, lines 1-10 and 21-24, page 5, lines 20-24, and page 11, lines 20-23 teaches that web browser based object components include HTML or web pages that provide web wizards.

Unlike Gauthier, dependent claim 2 requires, among other things, sub-wizard components that are web browser based object components. Gauthier fails to teach

browser based object components. Therefore, for at least the above reasons, Applicants respectfully request withdrawal of the anticipation rejection and allowance of claim 2.

**c. Dependent claim 3 requires, among other things, subwizard components that are operating system based application components.**

Claim 3 depends from claim 1 and further defines novel features of the claimed invention. Accordingly, claim 3 is allowable by virtue of its dependence on claim 1. Additionally, claim 3 is allowable because Gauthier fails to disclose “wherein one or more subwizard components are operating system based application component object extensions.” The Office has referenced Gauthier, at col. 6, ll. 42-46, as anticipating the claimed requirement. The section cited by the office action discloses:

“The operating system 122 provides the basic functionality that controls the computer system 100. Operating system 122 can comprise any suitable operating system, such as IBM's AS/400, OS/2, Microsoft's Windows, Java and the various flavors of UNIX”

In this section, Gauthier teaches the use of an operating system and the different types of operating systems. Gauthier, at col. 7, ll. 45-58, further discloses a wizard framework that defines the basic elements of a wizard. This framework defines the core functions of the solution, those elements that are required and cannot be extended by a developer. The framework also defines extensible functions of the solution, those that can be customized and extended by the developer. The customization/extension quality of framework mechanisms is extremely valuable because the cost of customizing or extending a framework is much less than the cost of replacing or reworking an existing solution.

Although Gauthier refers to extensions of particular functions, these types of extensions are limited specifically to objects within the wizard framework. There is no teaching of an extension that utilizes operating system based application components.

Applicants' specification, at pp. 14-15, illustrates that an operating system based extension comprises a object wizard component extension and is created in an operating environment that differs from the browser based object component which comprises the HTML component wizard extension.

Unlike Gauthier, dependent claim 3 requires, among other things, sub-wizard components that are operating system based application component object extensions. Gauthier fails to teach operating system based application component object extensions. Therefore, for at least the above reasons, Applicants respectfully request withdrawal of the anticipation rejection and allowance of claim 3.

**d. Independent claim 14 requires, among other things, chaining a first and second wizard via navigation components found on the first and second wizard.**

With respect to independent claim 14, Gauthier fails to teach, among other things, "at least one navigation component on each of said first and second wizards, said navigation components allowing sequential progression or regression through said first and second wizards to chain said second wizard to said first wizard to guide a user through the first and second tasks."

The Final Office Action has referenced Gauthier, at col. 9, l. 55-col. 10, l. 5, col. 10, ll. 57-67, and col. 14, ll. 9-14. Gauthier, at col. 10, ll. 20-30, teaches the WizardManagerButtonPanel class provides a panel with a plurality of buttons for use on the target wizard interface. These buttons would typically include standard GUI interface buttons, such as BACK, NEXT, FINISH, CANCEL and HELP. The fact that Gauthier uses the term subwizard does not mean navigation controls are maintained independently for each subwizard as claimed. Gauthier does not describe the identical invention of claim 14 because navigation buttons are not on **each wizard** as expressly required by independent claim 14. Instead, the buttons are centralized using a WizardManagerFrame, col. 10, ll. 5-40. Furthermore, as noted above the execution and control of

the subwizards of Gauthier are centralized; so, the subwizards are not chained through the use of separate navigational components. Gauthier, at col. 2, ll. 4-13 and col. 11, ll. 54-63, teaches that a wizard guides a user through a task and provides the ability to present GUI components that collect information from the user. However, Gauthier does not teach how to integrate two separate and distinct wizards in the manner claimed in independent claim 14. The reason for this is because Gauthier is not directed to linking two wizards: a host wizard and a subwizard as explained by Applicant specification at pp. 12-15.

Unlike Gauthier, independent claim 14 requires, among other things, at least one navigation component on each of said first and second wizards, said navigation components allowing sequential progression or regression through said first and second wizards to chain said second wizard to said first wizard to guide a user through the first and second tasks. Gauthier fails to teach navigation components on each wizard that allow sequential progression or regression through said first and second wizards to chain said second wizard to said first wizard to guide a user through the first and second tasks. Therefore, for at least the above reasons, Applicants respectfully request withdrawal of the anticipation rejection and allowance of independent claim 14.

#### **Rejections based on 35 U.S.C. § 103**

#### **IV. Applicable Authority**

Title 35 U.S.C. § 103(a) declares, a patent shall not issue when “the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains.” The Supreme Court in *Graham v. John Deere* counseled that an obviousness determination is made by

identifying: the scope and content of the prior art; the level of ordinary skill in the prior art; the differences between the claimed invention and prior art references; and secondary considerations. *Graham v. John Deere Co.*, 383 U.S. 1 (1966).

**V. Rejection under 35 U.S.C. § 103(a) over Fedorov (Professional Active Server Pages 2.0," 1998, Wrox Press Ltd) and Gauthier et al (U.S. Patent No. 6,574,791) should be withdrawn because Fedorov and Gauthier do not describe or suggest "passing a property bag between said host wizard and said one or more sub-wizard components."**

Claims 16, 18, and 19 stand rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,574,791 to Gauthier, et al. in view of Alex Fedorov, et al., "Professional Active Server Pages 2.0," 1998, Wrox Press Ltd ("Fedorov, et al.).

Dependent claims 16 and 18-19 further define novel features of the claimed embodiments and each depend either directly or indirectly, from one of independent claims 1, 8, and 9. Accordingly, for at least the reason set forth above with respect to independent claims 1, 8, and 9, dependent claims 16 and 18-19 are believed to be in condition for allowance by virtue of their dependency. 37 C.F.R. 1.75(c). As such, withdrawal of the 35 U.S.C. § 103(a) rejection of dependent claims 16 and 18-19 is respectfully requested.

Furthermore, with respect to claims 16 and 18-19, Gauthier and Fedorov, singularly or in combination fail to teach or suggest, among other things, "passing a property bag between said host wizard and said one or more sub-wizard components."

The Office concedes that Gauthier fails to disclose or suggest the claimed passing of a property bag. However, the Office contends that Fedorov remedies Gauthier's deficiency and teaches or suggests the claimed passing of a property bag. Applicants respectfully disagree. Fedorov, at page 423, expressly teaches that seismic.asp provides server-side calculations. Seismic.asp is not a host wizard. The Office contends that seismic.asp is the

claimed host wizard. But as set forth above seismic cannot be a host wizard. Fedorov, expressly discloses that Seismic.asp redirects the browser to the server when calculations are performed. Furthermore, Gauthier maintains centralized execution of the wizards so information does not need to be passed in manner described in claims 16 and 18-19.

Unlike Fedorov and Gauthier, singularly or in combination, the claimed embodiment requires passing a property bag between a host-wizard and other wizards when navigating and passing control among the wizards. Fedorov teaches utilizing a server to perform calculations on information collected by a wizard. There is nothing in Fedorov or Gauthier, singularly or in combination, that teaches or suggests passing a property bag between at least two wizards when navigating and transferring control between the wizards. Accordingly, for at least the foregoing reasons, Applicants respectfully request the withdrawal of the obviousness rejection and allowance of claims 16 and 18-19.

**New claims**

New claims 21 and 22 are added to round out the scope of protection and are allowable at least for the same reasons as independent claim 1.

**CONCLUSION**

For at least the reasons stated above, the pending claims are now in condition for allowance. Applicants respectfully request withdrawal of the pending rejections and allowance of the claims. If any issues remain that would prevent issuance of this application, the Examiner is urged to contact the undersigned to resolve the same. It is believed that no fee is due, however, the Commissioner is hereby authorized to charge any amount required to Deposit Account No. 19-2112 referencing 164122.01/MFCP.88142.

Respectfully submitted,

/ MONPLAISIR HAMILTON/

MONPLAISIR HAMILTON  
Reg. No. 54,851

MGHZ/jc  
SHOOK, HARDY & BACON L.L.P.  
2555 Grand Blvd.  
Kansas City, MO 64108-2613  
816-474-6550